Data Analysis and Visualization in R (IN2339)

Exercise Session 4 - Low dimensional visualization

Daniela Klaproth-Andrade, Jun Cheng, Daniel Bader, Julien Gagneur

Section 00 - Getting ready

1. Make sure you have already installed and loaded the following libraries:

```r
library(ggplot2)
library(data.table)
library(magrittr)   # Needed for %>% operator
library(tidyr)

library(MAS6005)  # Install with devtools::install_github("OakleyJ/MAS6005")
library(ggrepel)
```

Section 01 - Plot types

Match each chart type with the relationship it shows best.

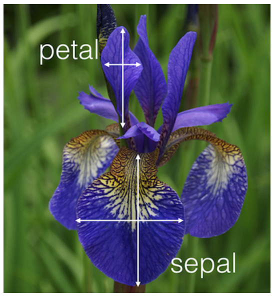
1. shows distribution and quantiles, especially useful when comparing distributions.
2. highlights individual values, supports comparison and can show rankings or deviations categories and totals
3. shows overall changes and patterns, usually over intervals of time
4. shows relationship between two continues variables.

Options: bar chart, line chart, scatterplot, boxplot

```r
# 1. boxplot
# 2. bar chart
# 3. line chart
# 4. scatterplot
```

Section 02 - Visualizing distributions

`Iris` is a classical dataset in machine learning literature. It was first introduced by R.A. Fisher in his 1936 paper.



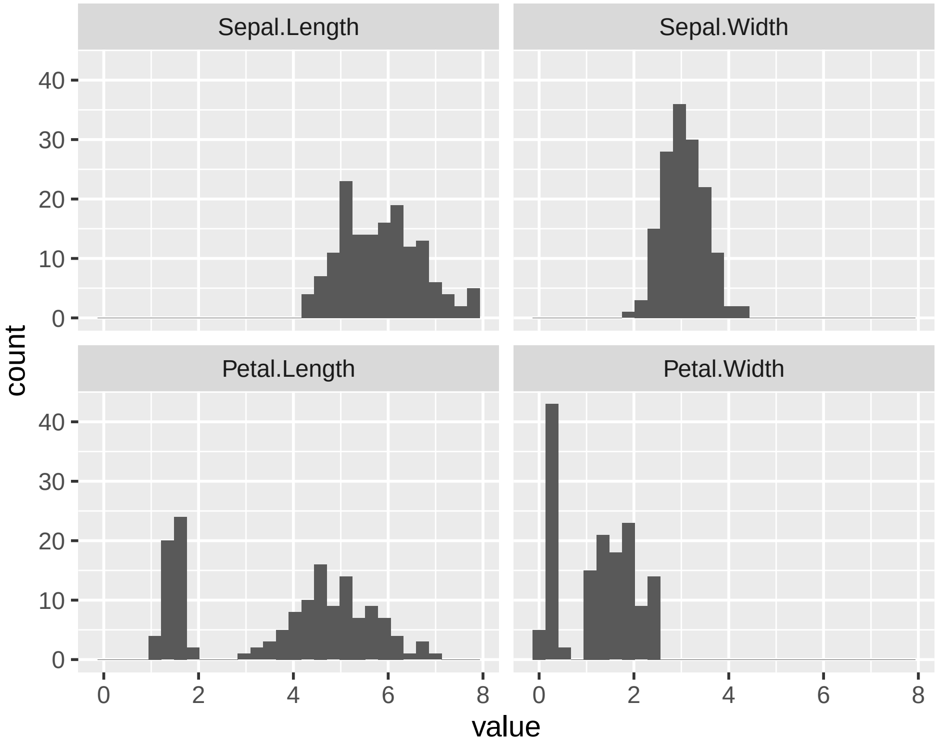
1. Load the *iris* data and transform it to a `data.table`. Have a look at its first and last rows.

```
iris <- as.data.table(iris)
iris
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
##   1:          5.1         3.5          1.4         0.2    setosa
##   2:          4.9         3.0          1.4         0.2    setosa
##   3:          4.7         3.2          1.3         0.2    setosa
##   4:          4.6         3.1          1.5         0.2    setosa
##   5:          5.0         3.6          1.4         0.2    setosa
##  ---                                                            
## 146:          6.7         3.0          5.2         2.3 virginica
## 147:          6.3         2.5          5.0         1.9 virginica
## 148:          6.5         3.0          5.2         2.0 virginica
## 149:          6.2         3.4          5.4         2.3 virginica
## 150:          5.9         3.0          5.1         1.8 virginica
```

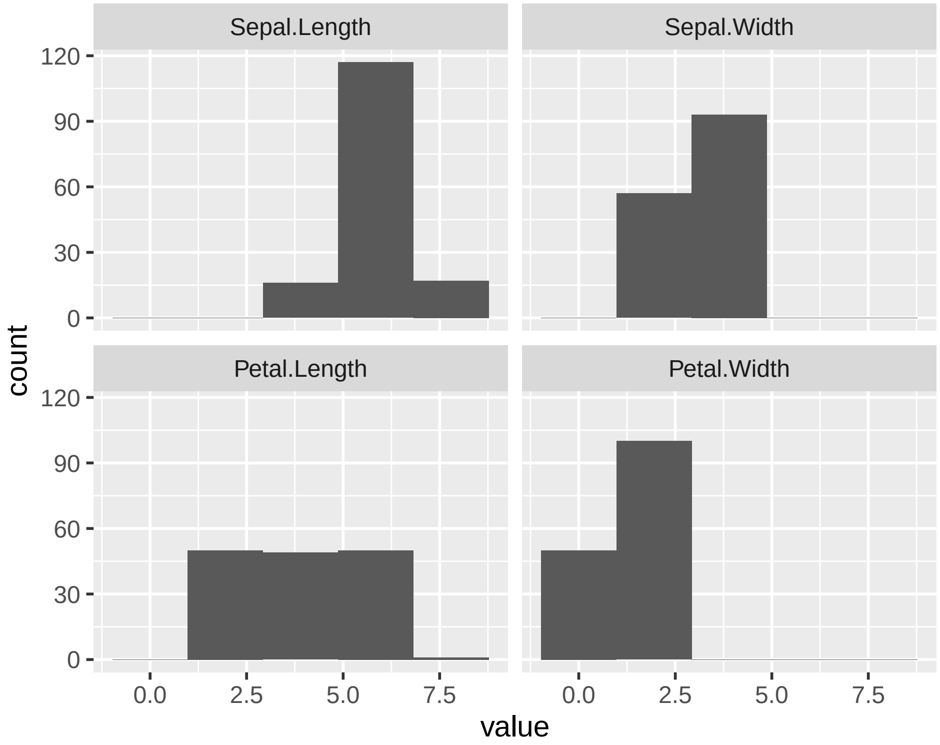
2. How are the lengths and widths of sepals and petals distributed? Make one plot of the distributions with multiple facets. *Hint:* You will need to reshape your data so that the different measurements (petal length, sepal length, etc.) are in one column and the values in another. Remember which is the best plot for visualizing distributions.

```
# Solution
iris_melt <- melt(iris, id.var=c("Species"))
iris_melt %>%
  ggplot(aes(value)) +
  geom_histogram() +
  facet_wrap(~variable)
```

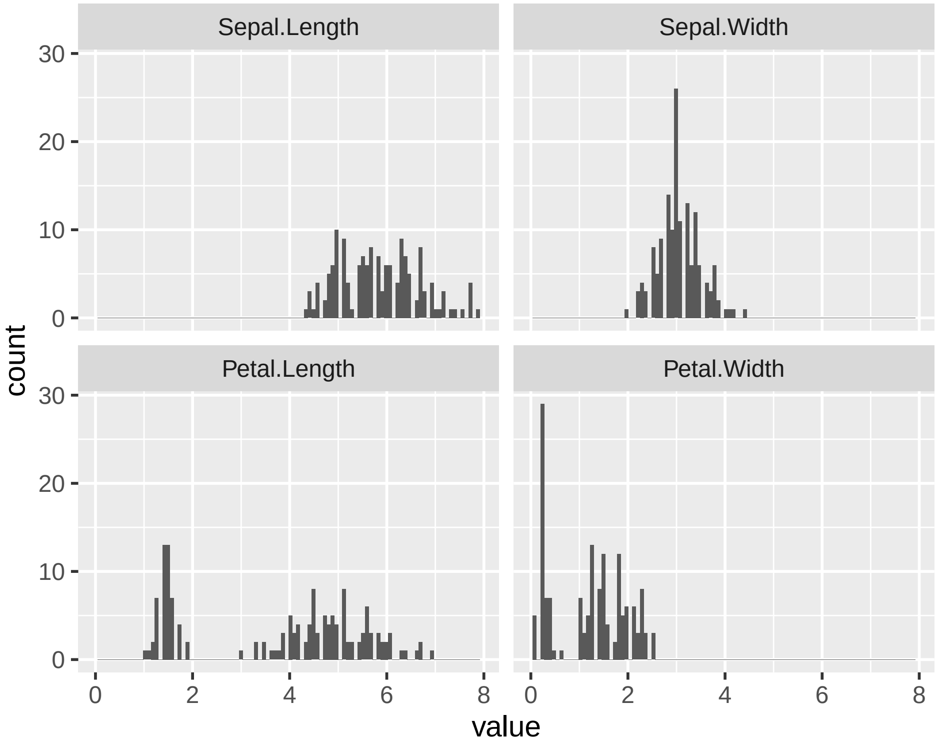


3. Vary the number of bins in the created histogram. Describe what you see.

```
## With very few bins, we cannot show the bimodal distribution correctly.
iris_melt %>%
  ggplot(aes(value)) +
  geom_histogram(bins=5) +
  facet_wrap(~variable)
```

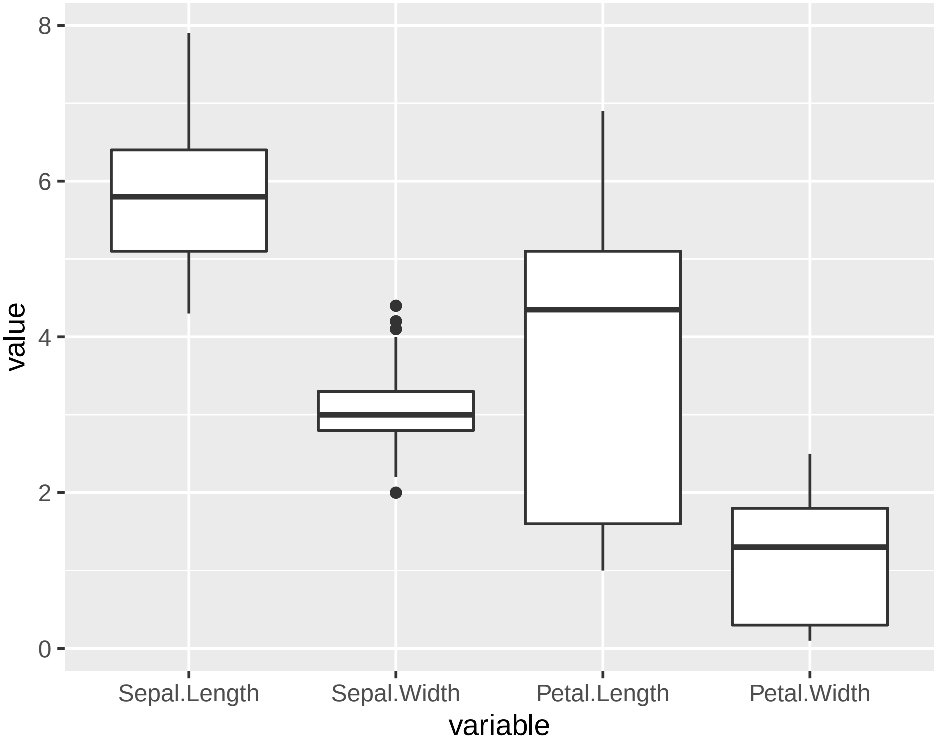


```
## With too many bins, the plot looks spiky
iris_melt %>%
  ggplot(aes(value)) +
  geom_histogram(bins=100) +
  facet_wrap(~variable)
```



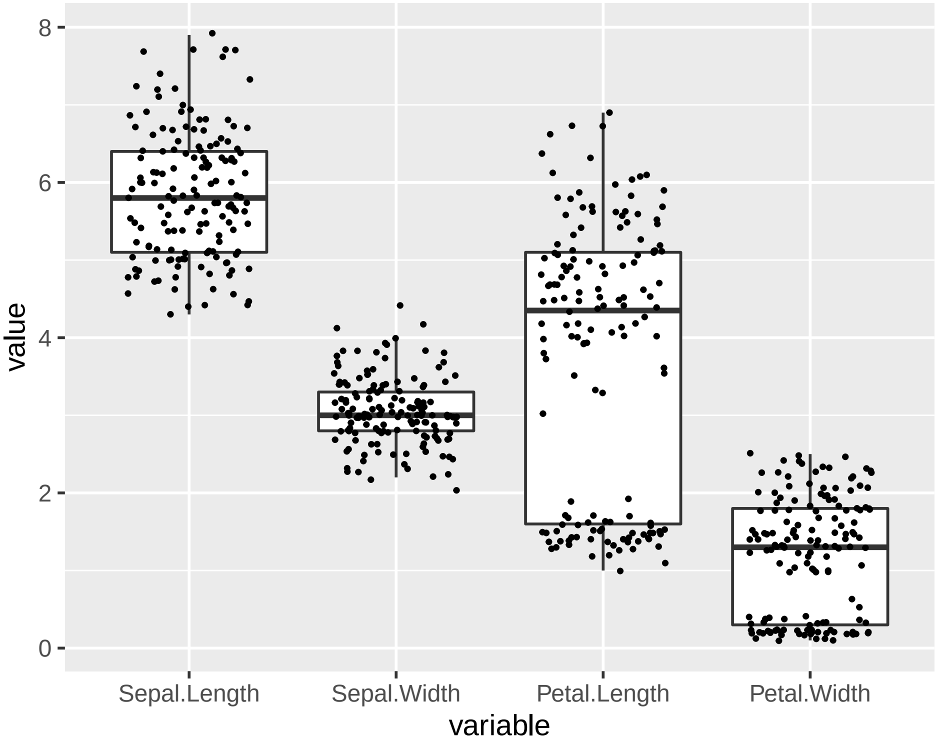
4. Visualize the lengths and widths of the sepals and petals from the iris data with boxplots.

```
ggplot(iris_melt, aes(variable, value)) +
   geom_boxplot()
```

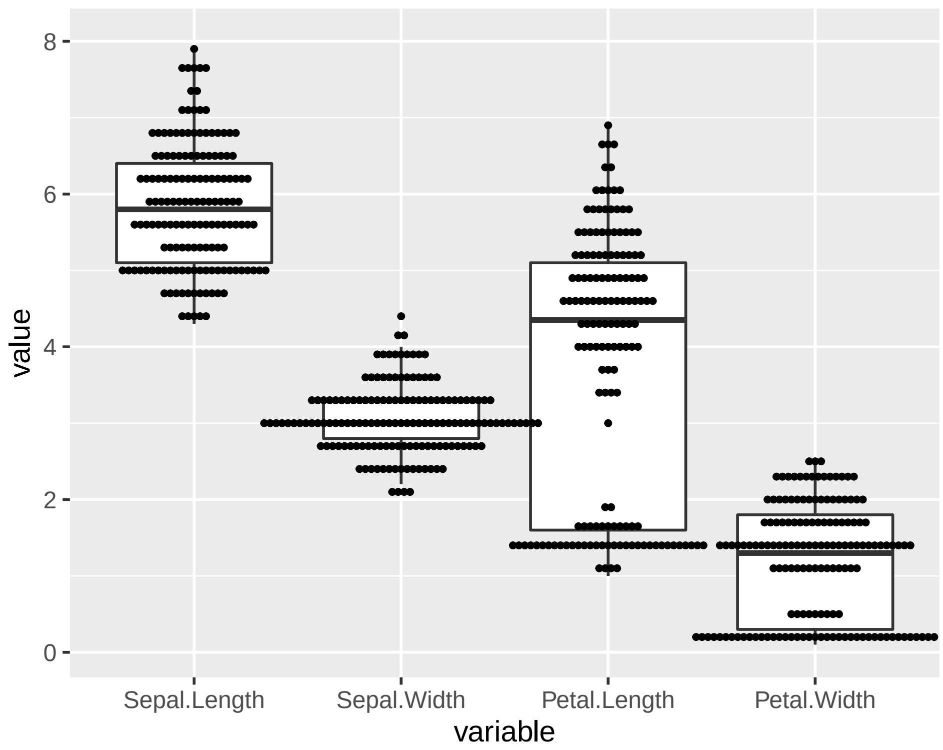


5. Add individual data points as dots on the boxplots to visualize all points. Discuss: in this case, why is it not good to visualize the data with boxplots? *Hint:* `geom_jitter()`

```
# petal distributions are bimodal, boxplot cannot visualize this property.
p <- ggplot(iris_melt, aes(variable, value)) +
      geom_boxplot(outlier.shape = NA)

p + geom_jitter(width = 0.3, size = .5)
```

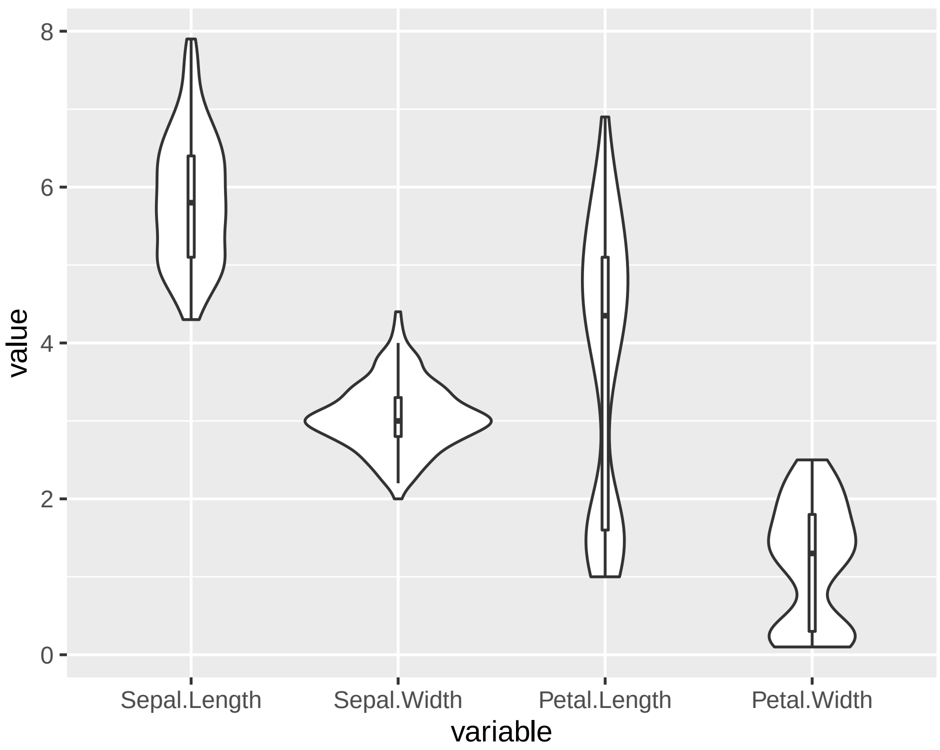


```
p + geom_dotplot(binaxis="y", stackdir="center", dotsize=0.3)
```



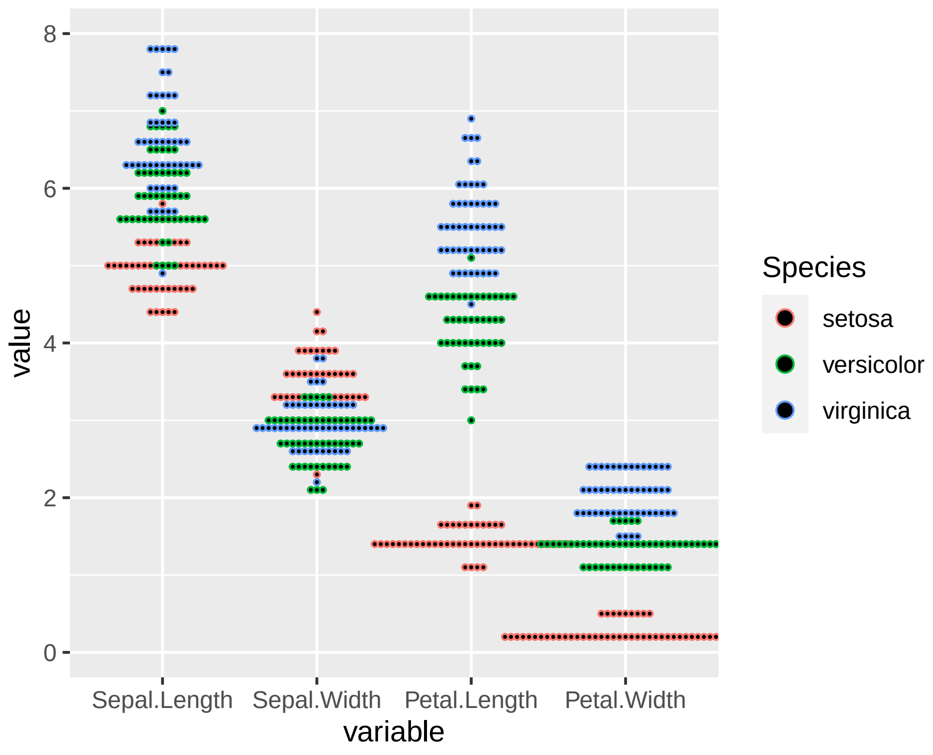
6. Alternatives to boxplot are violin plots (`geom_violin()`). Try combining a boxplot with a violinplot to show the the lengths and widths of the sepals and petals from the iris data.

```
ggplot(iris_melt, aes(variable, value)) +
  geom_violin() +
  geom_boxplot(width=0.03, outlier.shape=NA) # Overlay boxplot to visualize median and IQR.
```



7. Which pattern shows up when moving from boxplot to a violin plot? Investigate the dataset to explain this kind of pattern, provide with visualization.

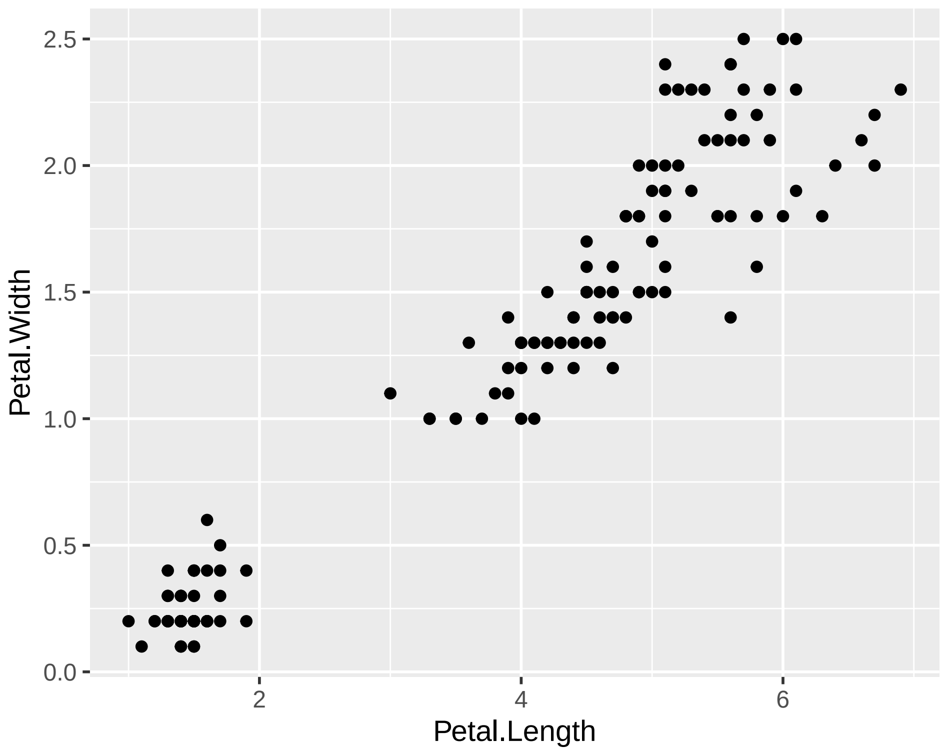
```r
# We see that petal length and petal width are bimodal.
# As the iris data set has 3 species, the different belong
# to the different species, so we can color the dots by Species.
ggplot(iris_melt, aes(variable, value, color = Species)) +
  geom_dotplot(binaxis="y", stackdir="centerwhole", dotsize=0.3)
```



Section 03 - Visualizing relationships

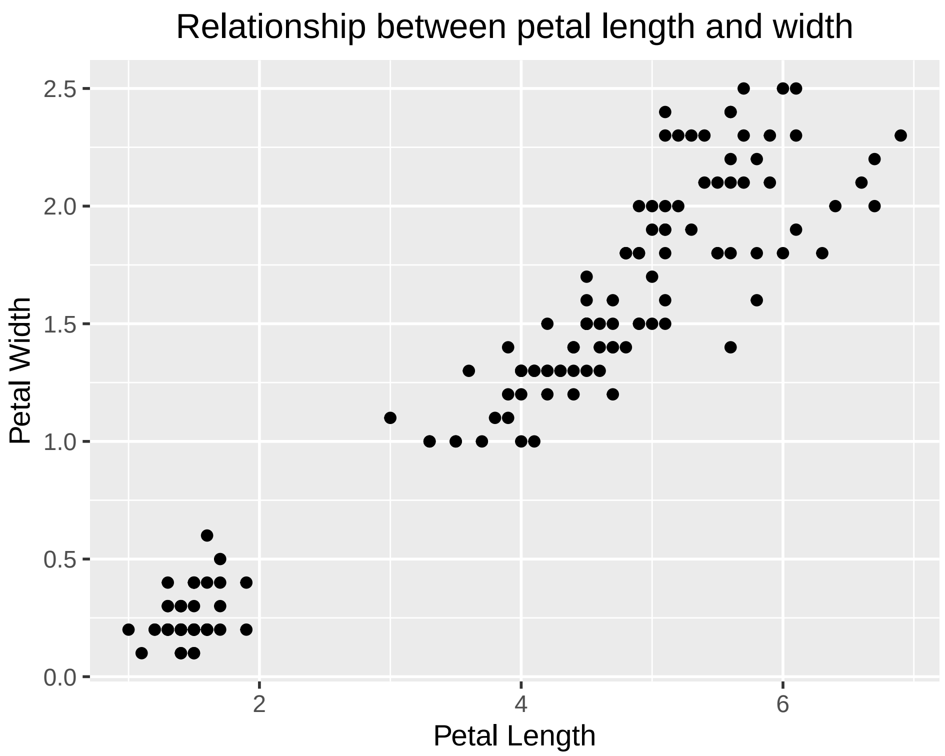
1. Are there any relationships/correlations between petal length and width? How would you show it?

```
# Yes, they correlate. We use a scatter plot for showing this:
ggplot(iris,aes(Petal.Length,Petal.Width)) +
   geom_point()
```



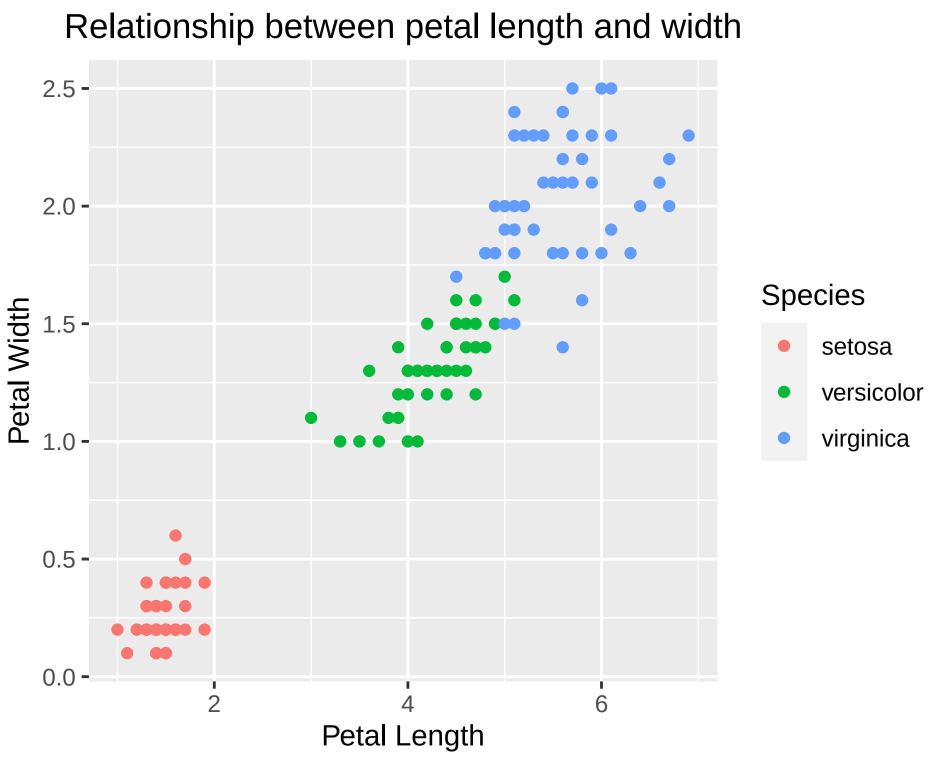
2. [OPTIONAL] Change your plot title and axis labels in the previous plot. For instance, the new title can be "Relationship between petal length and width", and the axis labels "Petal Length" and "Petal Width", respectively.

```
ggplot(iris,aes(Petal.Length,Petal.Width)) +
 geom_point() +
 labs(x = "Petal Length", y = "Petal Width",
      title = "Relationship between petal length and width") +
 theme(plot.title = element_text(hjust=0.5))
```

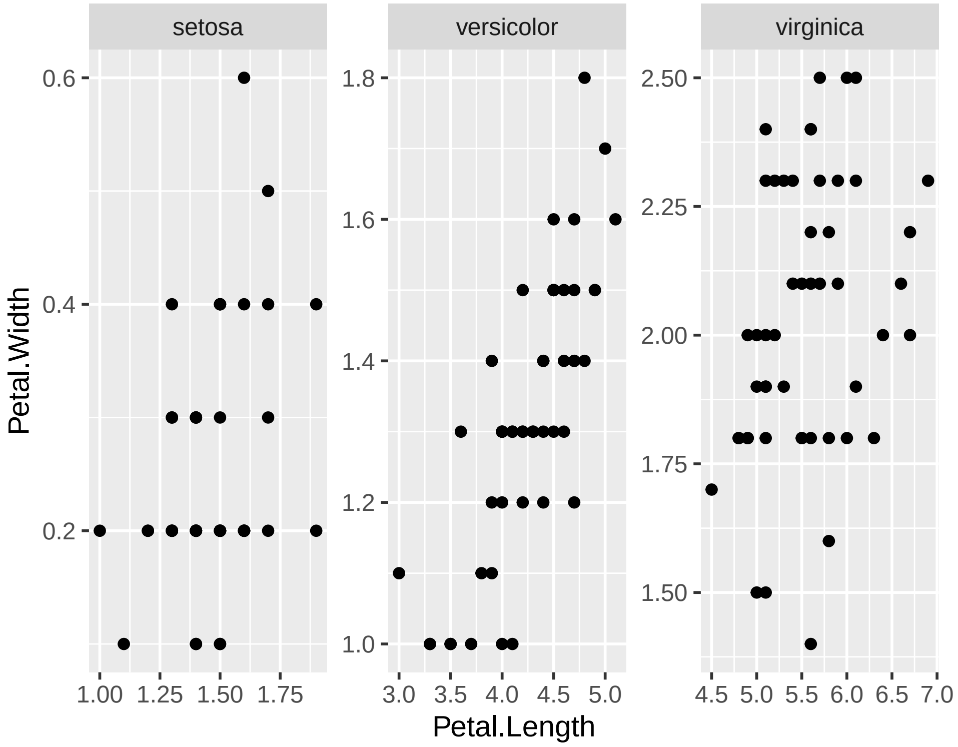


3. Do petal lengths and widths correlate in every species? Show this with a plot.

```
# They correlate on every species, add color or facets with respect to ‘Species‘

## With coloring
ggplot(iris,aes(Petal.Length,Petal.Width, color=Species)) +
  geom_point() +
  labs(x = "Petal Length", y = "Petal Width",
       title = "Relationship between petal length and width") +
  theme(plot.title = element_text(hjust=0.5))
```



```
# With facets
ggplot(iris,aes(Petal.Length,Petal.Width)) +
  geom_point() +
  facet_wrap(~Species,  scales = 'free')
```



```
# scales = ’free’, relax axis in each plot to fit its own data.
```

Section 04 - The importance of data visualization

Anscombe’s quartet was constructed in 1973 by the statistician Francis Anscombe to demonstrate both the importance of graphing data before analyzing it, and the effect of outliers on statistical properties. `anscombe` is directly built in R. You don’t need to load it.

1. We reshaped the original `anscombe` data to `anscombe_reshaped`. Which one is tidier?

```
anscombe_reshaped <- anscombe %>%
  as.data.table %>%
  .[, ID := seq(nrow(.))] %>%
  melt(id.var=c("ID")) %>%
  separate(variable, c(’xy’, "group"), sep=1) %>%
  dcast(... ~ xy) %>%
  .[, group := paste0("dataset_", group)]
```

```
# anscombe is not tidy because the column names correspond to variables x and y
# anscombe_reshaped is the tidy version
```

2. Compute the mean and standard deviation of each variable for each group. What do you see?

```
# Use the functions ‘mean()‘ and ‘sd()‘ and create new columns
anscombe_reshaped[, .(x_mean = mean(x),
                      y_mean = mean(y),
                      x_sd = sd(x),
                      y_sd = sd(y)),
                  by = "group"]
```

```
##        group x_mean   y_mean     x_sd     y_sd
## 1: dataset_1      9 7.500909 3.316625 2.031568
## 2: dataset_2      9 7.500909 3.316625 2.031657
## 3: dataset_3      9 7.500000 3.316625 2.030424
## 4: dataset_4      9 7.500909 3.316625 2.030579
```

3. For each dataset, what is the Pearson correlation between x and y? *Hint:* `cor()` and Wikipedia[1] for Pearson correlation.

```
# Group by ‘group‘ and use the function ‘cor()‘
anscombe_reshaped[, .(correlation = cor(x, y)), by = ’group’]
```

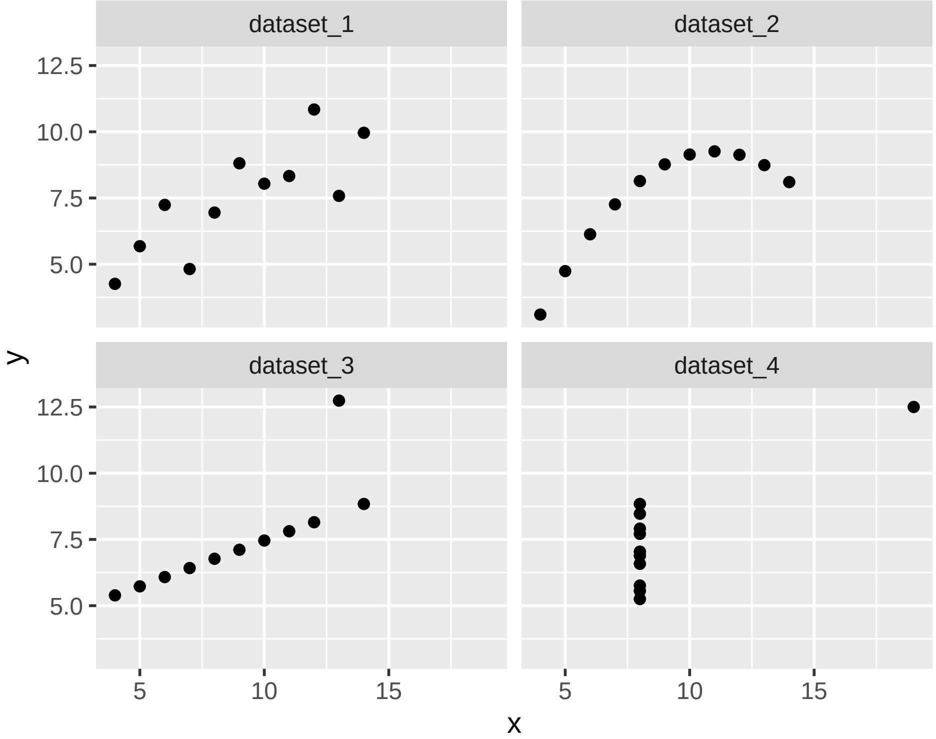
```
##        group correlation
## 1: dataset_1   0.8164205
## 2: dataset_2   0.8162365
## 3: dataset_3   0.8162867
## 4: dataset_4   0.8165214
```

4. Only by computing statistics, we could conclude that all 4 datasets have the same data. Now, plot x and y for each dataset and discuss.

```
# It’s always important to plot the raw data!
# Different distributions can have the same mean and sd.
ggplot(anscombe_reshaped, aes(x, y)) +
```

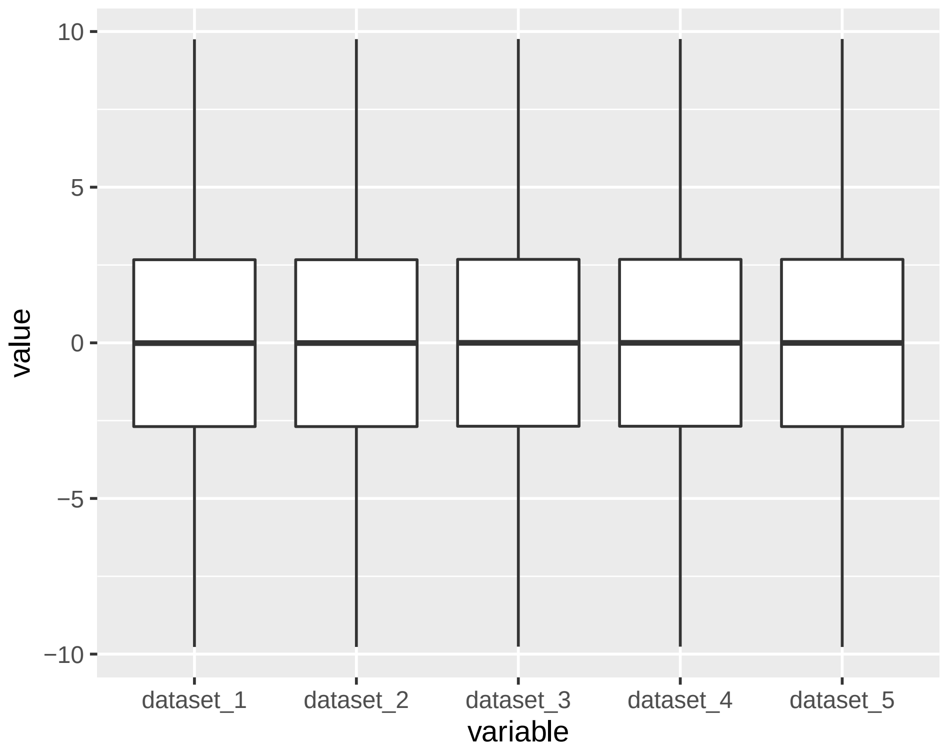
[1] https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

```
  geom_point() +
  facet_wrap(~ group)
```



5. [OPTIONAL] Consider now the datasets given in the file `boxplots.csv`. Load the data and visualize the different datasets with a boxplot. What do you see? What can you conclude?

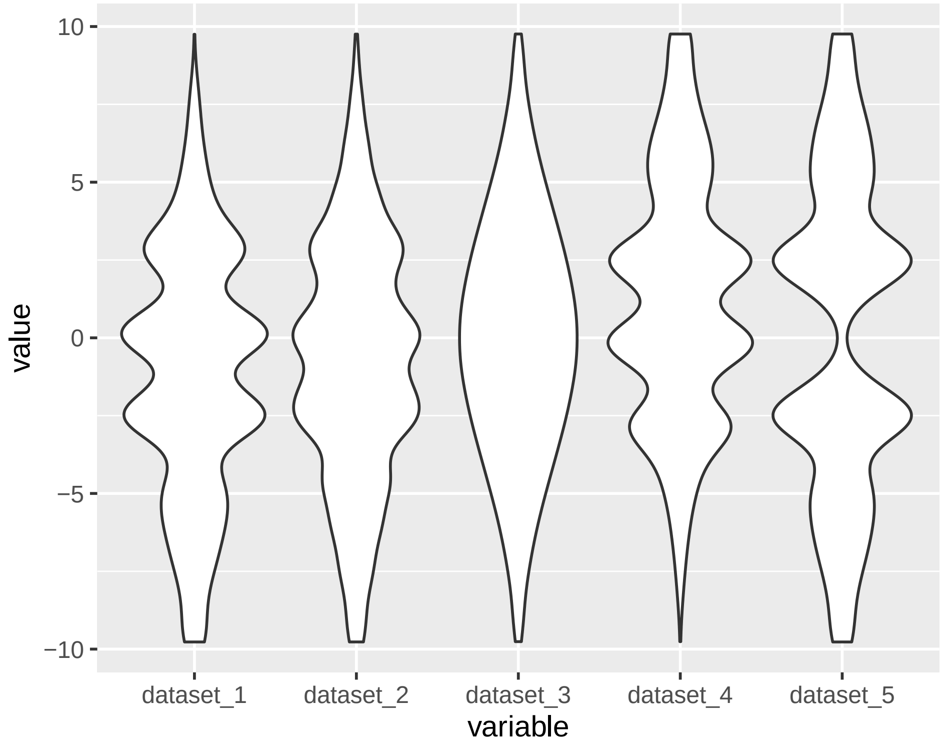
```
boxplots_dt <- fread('extdata/boxplots.csv')
melt(boxplots_dt) %>% ggplot(aes(variable, value)) + geom_boxplot()
```



```
## From the boxplots we conclude that all datasets have the same statistics (standard dev, median, IQR)
## But are the identical?
```

6. [OPTIONAL] Exchange the boxplots by violin plots in the previous exercise. Did something change? What do you conclude?

```
melt(boxplots_dt) %>% ggplot(aes(variable, value)) + geom_violin()
```



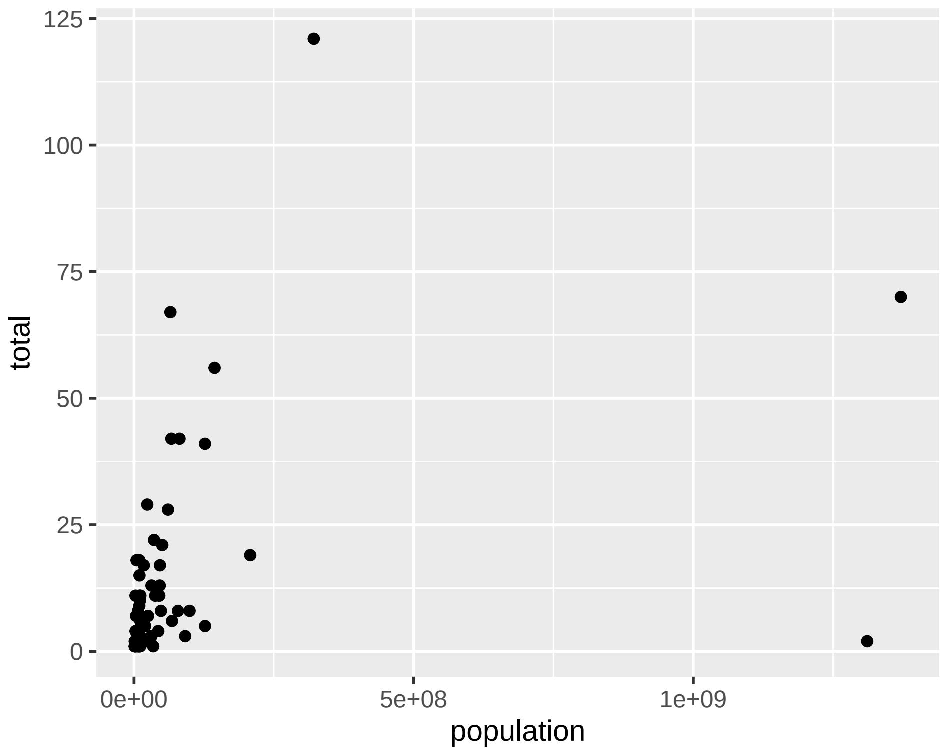
```
## The datasets are not identical. This is shown by the violin plots and not by the boxplots
## Boxplots not always helpful!! (violin plots also not always helpful)
```

Section 05 - Axes scaling and text labeling

1. Consider the `medals` dataset from the `MAS6005` library. Compare total number of medals won against population size in the 2016 Rio olympics with a scatter plot. You can load the dataset with the following code:

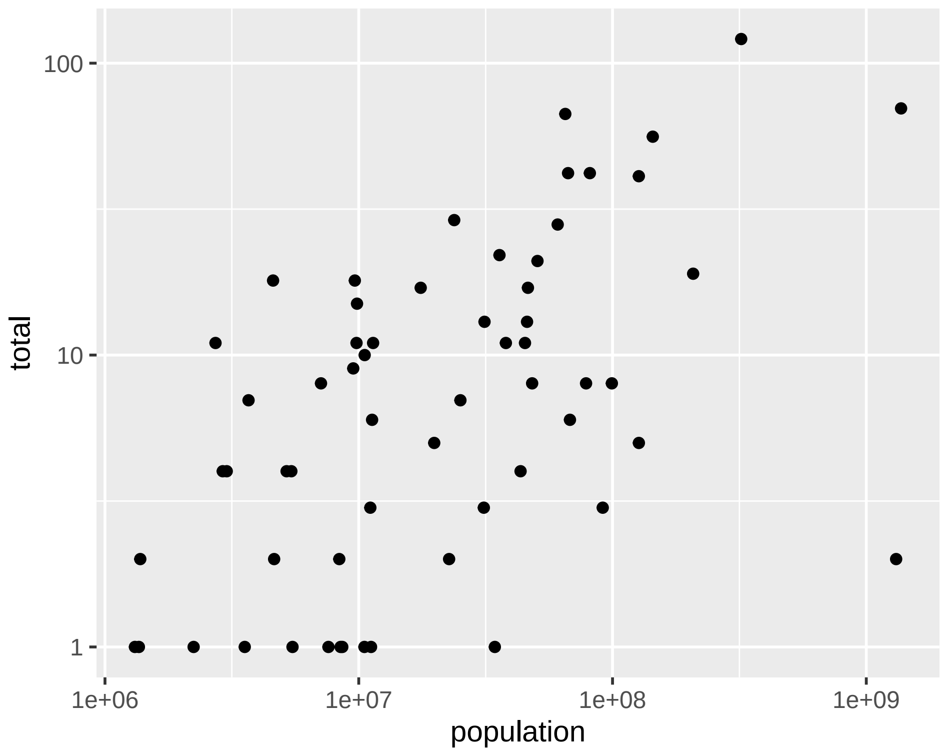
```
library(MAS6005)
attach(medals)
medals_dt <- as.data.table(medals)
```

```
ggplot(medals_dt, aes(population, total)) + geom_point()
```



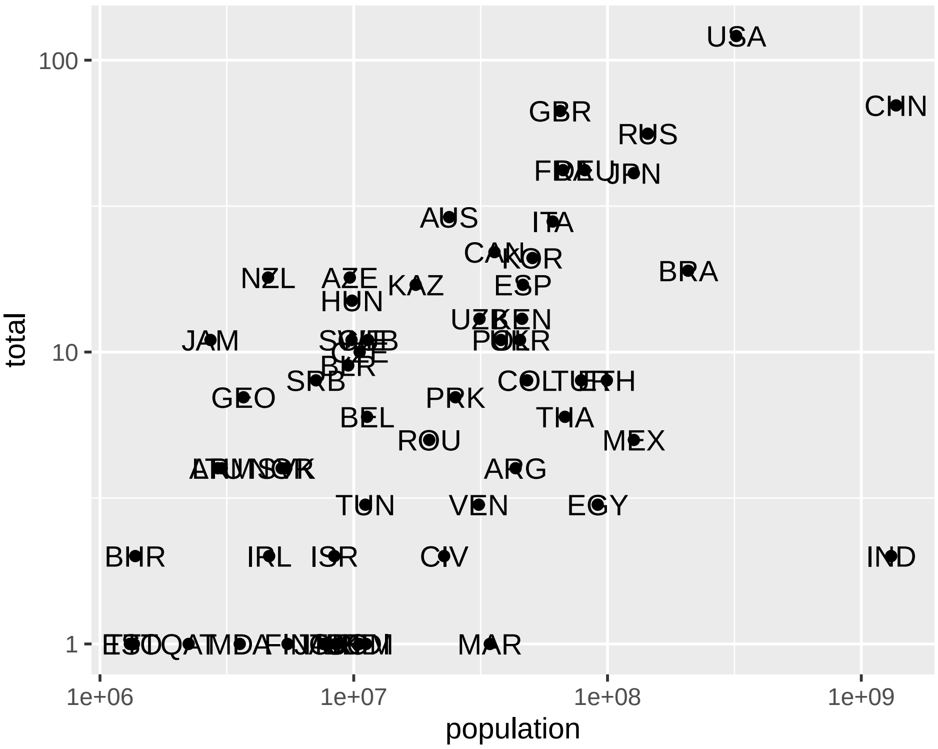
2. What are the problems with the previous plot? Solve these issues with an adapted version of the plot.

```r
# Problem:
# There are two countries with much larger populations than the rest.
# This 'distorts' the plot somewhat, in that a lot of the remaining points are bunched together.

# Solution: log scaling
ggplot(medals_dt, aes(population, total)) + geom_point() + scale_x_log10() + scale_y_log10()
```

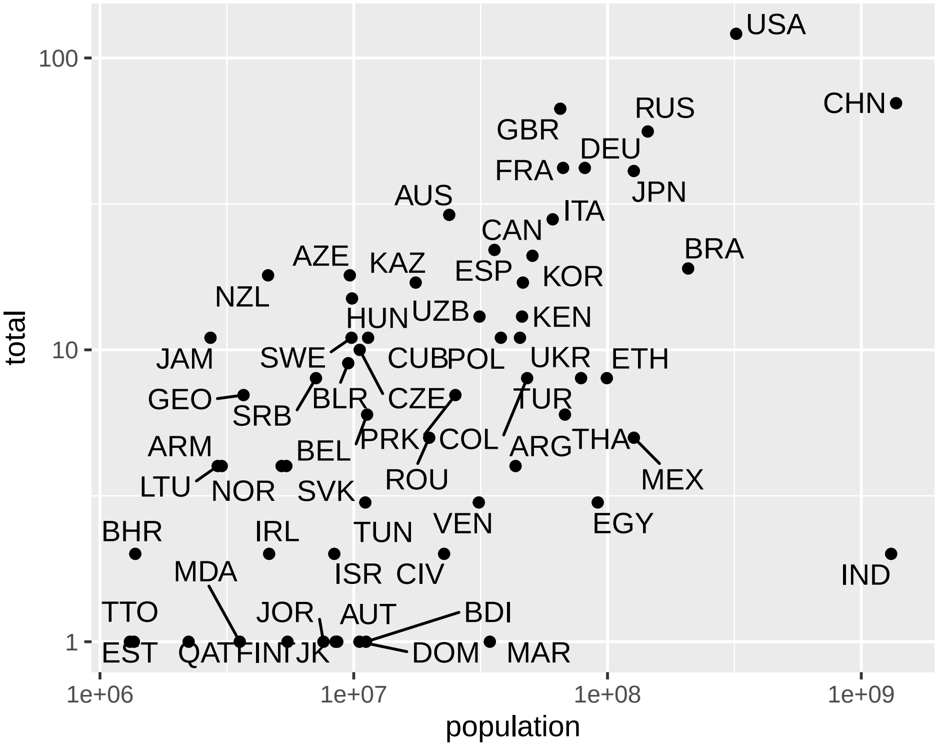


3. Add the country labels to the points in the scatter plot. Compare the differences of using the library `ggplot2` and the library `ggrepel` for this task

```
# Overlapping labels
ggplot(medals_dt, aes(population, total)) + geom_point() + scale_x_log10() + scale_y_log10() +
  geom_text(aes(label=code))
```



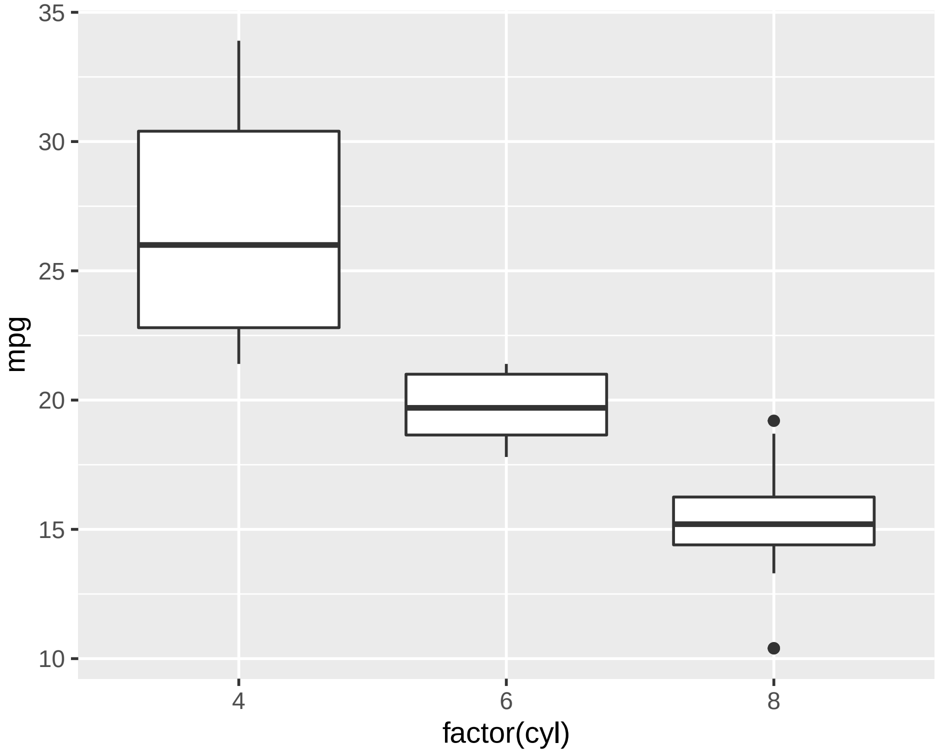
```r
# Non-overlapping labels with ggrepel
library(ggrepel)
ggplot(medals_dt, aes(population, total)) + geom_point() + scale_x_log10() + scale_y_log10() +
  geom_text_repel(aes(label=code))
```



Section 06 - Understanding and recreating boxplots

1. [OPTIONAL] Using the `mtcars` dataset, make a boxplot of the miles per gallon (mpg) per cylinder (cyl).

```
mtcars <- data.table(mtcars)
ggplot(mtcars, aes(factor(cyl), mpg)) + geom_boxplot()
```

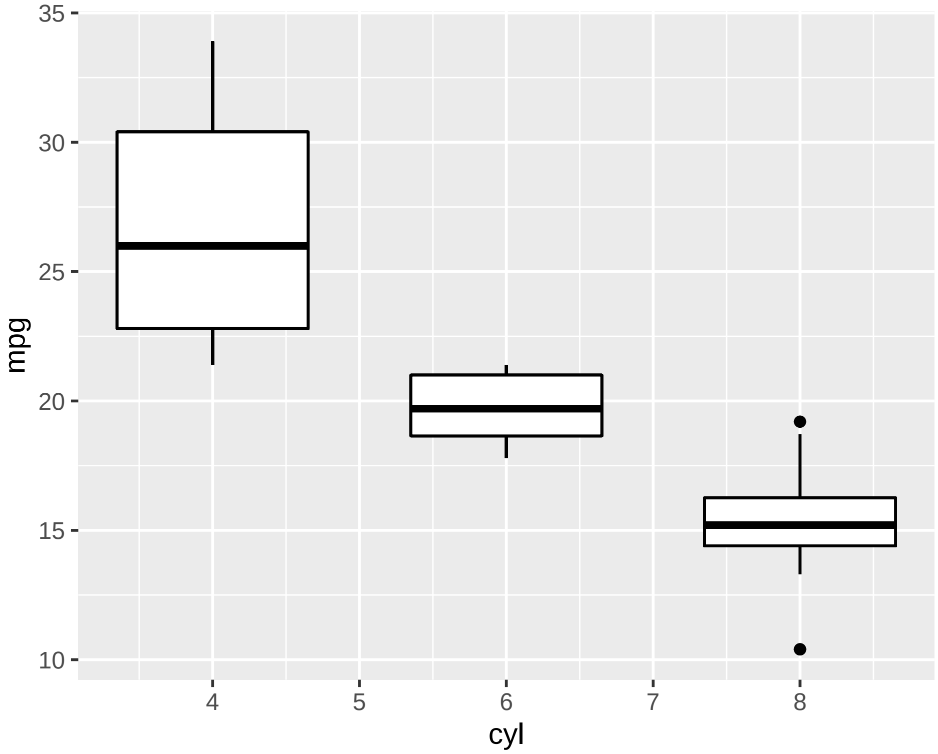


2. [OPTIONAL] Now, recreate the same plot without using `geom_boxplot`. You have to add all the layers manually: IQR box, median line, whiskers and outlier points. *Hint*: Remember how a boxplot is constructed[2]. You may find these functions useful: `IQR`, `geom_crossbar`, `geom_segment`, `geom_point`. Use `data.table` commands.

```
## First compute median
mtcars[, medians := median(mpg), by = cyl]

## Quantiles
mtcars[, c("lq", "uq") := .(quantile(mpg, 0.25), quantile(mpg, 0.75)), by = cyl]

## Whiskers
mtcars[, IQR := 1.5 * IQR(mpg), by = cyl]
mtcars[, c("up_IQR", "down_IQR") := .(IQR + uq, lq - IQR)]

## Get the most extreme value within 1.5*IQR
mtcars[mpg < up_IQR,
       up_whisker := max(mpg),
             by = "cyl"]
mtcars[mpg > down_IQR,
       down_whisker := min(mpg),
               by = "cyl"]
```

[2] http://docs.ggplot2.org/current/geom_boxplot.html

```
## Compute outliers
mtcars[, outlier := (mpg < down_IQR | mpg > up_IQR), by = 'cyl']

## Make the plot
ggplot(mtcars, aes(cyl, medians, ymax = uq, ymin = lq)) +
  geom_crossbar(fill = 'white', width = 1.3) +
  geom_segment(aes(cyl, down_whisker, xend = cyl, yend = lq)) +
  geom_segment(aes(cyl, uq, xend = cyl, yend = up_whisker)) +
  geom_point(data = mtcars[outlier == TRUE], aes(cyl, mpg)) +
  labs(y = "mpg")
```



APPENDIX - Solutions to the quizes from the lecture

Quiz

Calculating the mean height returns the following output:

```
height_dt[, mean(height)]
```

What happened?

1. `mean()` is not the right function to assess what we want to know.
2. Adults in Germany are exceptionally tall
3. A decimal point error in one data point.
4. It's a multiple testing problem because we are looking at so many data points (n=100).

Solution

What happened?

- 1. `mean()` is not the right function to assess what we want to know.
  - *No, the mean is exactly what we want.*
- 2. Adults in Germany are exceptionally tall.
  - *OK, no…*
- **3. A decimal point error in one data point.**
  - *Yes, see next slide.*
- 4. It's a multiple testing problem because we are looking at so many data points (n=100).
  - *This question was intentionally misleading, this does not have anything to do with multiple testing.*

Quiz

When to use a line plot?

1. To show a connection between a series of individual data points
2. To show a correlation between two quantitative variables
3. To highlight individual quantitative values per category
4. To compare distributions of quantitative values across categories

Solution

When to use a line plot?

1. **To show a connection between a series of individual data points**

A line plot can be considered for connecting a series of individual data points or to display the trend of a series of data points. This can be particularly useful to show the shape of data as it flows and changes from point to point.

Quiz

What's the result of the following command?

```
ggplot(data = mpg)
```

1. Nothing happens
2. A blank figure will be produced
3. A blank figure with axes will be produced
4. All data in `mpg` will be visualized

Solution

`ggplot(data = mpg)`: a blank figure will be produced

Quiz

What's the result of the following command?

```
ggplot(data = mpg, aes(x = hwy, y = cty))
```

1. Nothing happens
2. A blank figure will be produced
3. A blank figure with axes will be produced
4. A scatter plot will be produced

Solution

`ggplot(data = mpg, aes(x = hwy, y = cty))`: A blank figure with axes will be produced

Quiz

What's the result of the following command?

```
ggplot(data = mpg, aes(x = hwy, y = cty)) + geom_point()
```

1. Nothing happens
2. A blank figure will be produced
3. A blank figure with axes will be produced
4. A scatter plot will be produced

Solution

`ggplot(data = mpg, aes(x = hwy, y = cty)) + geom_point()`: A scatter plot will be produced

Quiz

For which type of data will boxplots produce meaningful visualizations? (2 possible answers)

1. For discrete data.
2. For bi-modal distributions.
3. For non-Gaussian, symmetric data.
4. For exponentially distributed data.

Solution

For which type of data will boxplots produce meaningful visualizations?

- **3. For non-Gaussian, symmetric data.**
- **4. For exponentially distributed data.**

Boxplots are bad for bimodal data since they only show one mode (the median), but are ok for both symmetric and non-symmetric data, since the quartiles are not symmetric.